

AUSTRALIAN OS9 NEWSLETTER

Volume 8

July 1994

Number 5

EDITOR:	Gordon Bentzen	(07) 344-3881
SUB-EDITOR:	Bob Devries	(07) 278-7209
TREASURER:	Jean-Pierre Jacquet	(07) 372-4675
	Fax messages	(07) 372-8325
LIBRARIAN:	Bob Devries	
CONSULTANT:	Don Berrie	(079) 75-3537
SUPPORT:	Brisbane OS9 Users Group	

CONTENTS

OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9
OS9									OS9
OS9	Editorial	Page 2							OS9
OS9	INFO! by Rod Holden	Page 2							OS9
OS9	More on IBM - CoCo emulator ..	Page 4							OS9
OS9	Compressor extensions.....	Page 5							OS9
OS9	MM/1 Keysense function	Page 5							OS9
OS9	Record locking	Page 6							OS9
OS9	SCANF discussion	Page 7							OS9
OS9	Copyright discussion	Page 10							OS9
OS9	More INFO!	Page 14							OS9
OS9									OS9
OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9

EDITORIAL MATERIAL
Gordon Bentzen
8 Odin Street
SUNNYBANK QLD 4109

LIBRARY REQUESTS
Bob Devries
21 Virgo Street
INALA QLD 4077

AUSTRALIAN OS9 NEWSLETTER
Newsletter of the National OS9 User Group
Volume 8 Number 5

EDITOR : Gordon Bentzen
SUBEDITOR : Bob Devries

SUPPORT : Brisbane OS9 Level 2 Users Group.

TREASURER : Jean-Pierre Jacquet
LIBRARIAN : Rod Holden

Here we are, next to last. By that I mean that our last newsletter is due to go in the month of August this year.

Although the monthly newsletter will cease, that does not mean that we are all going to drop OS-9. A number of us plan to continue our interest in OS-9, OSK and OS9000, plus we are expecting a significant amount of P.D. software from our international connections.

But more on that next month by Bob Devries.

Also in our final edition next month we plan to include a current membership listing with postal addresses, so please contact myself or Bob Devries if you do not wish your details to be included.

In this edition we have included some discussion material from the Internet messages which are of interest, so without taking up a lot of space this time I will wish you all happy computing until next month.
Gordon.

IIIIII	NNNN	NN	FFFFFF	000000	!!
II	NN NN	NN	FF	00 00	!!
II	NN NN	NN	FFFF	00 00	!!
II	NN NN	NN	FF	00 00	!!
II	NN NN	NN	FF	00 00	!!
IIIIII	NN	NNNN	FF	000000	00

By Rod Holden

Hi, and welcome to Info. This particular piece of software is in a file called UltraDir.pak which is available under the OCN OS9_UTI directory, please read on;

Dir Documentation

This utility is not your run of the mill directory utility, it combines a lot of features into one routine and is a must for all serious OS9 user's. It is similar to the 68000 system utility but with one enhancement, the file searching can be directed to a specific directory.

Features:

1. Directory sorting.
2. Organizing directories in sorted order, permanently.
3. Multiple listings on one command line or wild cards.
4. File searches.
5. File searches, with wild cards * ? supported by Shell+ v2.1

6. Recursive directory listings.
7. Recursive file searches
8. Unformatted output for piping to filters.
9. Unformatted output with file matching.
10. Recursive levels upto 99.
11. Parameter placement and order not important.
12. Uses the OSK parameter delimiters of "-"
13. Built in helps.
14. Dynamic buffer allocation.

The file dir.bin is to be placed in your CMDS directory as "dir" and is a replacement for the old utility.

Help Display:

To display the help menu type the following, because of wild card usage I had to switch to a "h" instead of the "?" shell+ doesn't respect the "-" delimiters:

Example: dir -h produces the following message:

use: dir [opts] {dir1 dir2 ..} [{file name list}] [opts]

AUSTRALIAN OS9 NEWSLETTER

-e Extended directory listing
-x Execution directory listing
-s Sorted listing, alpha numeric order
-u Unformatted listing
-w Re-Write sorted directory
-h Help
-n<=dir> Search names, <in named directory>, optional
-r<=num> Recursive listing of directories, <num=level>, optional

Sorted Listings:

All operations can be displayed in sorted order by using the "-s" parameter.

Examples:

dir -s or dir -es or dir -esr or dir -enx d* -s

Multiple Directories:

More than one directory listing can be displayed by typing more than one Directory Name on the command line or using wild cards.

Examples:

dir SYS DEFS LIB

Lists the contents of the SYS, DEFS and LIB directories.

or dir -s S*

Lists the contents in sorted order of the SYS directory and SYSSRC, "my system specific".

File Searching:

Specific files can be named on the command line by using wild cards * ?, or by typing the specific name. Searches can be accomplished for the whole disk or directory specified by using the recursive parameter "-r". Listings can be output as formatted/unformatted or as an extended listing.

Examples:

dir -xens copy dir

Searches the CMDS directory for the files "dir" and "copy" and displays if found in sorted extended order.

dir -nes dir.doc copy.doc

Searches for the file "dir.doc" and "copy.doc" in the current "." directory.

Note: =<dir> the "<dir>" is an optional parameter the default is the current working directory.

dir -n=SOURCES/ASM dir.a copy.a -es

Searches the directory SOURCES/ASM for the files "dir.a" and "copy.a" and displays in extended sorted order.

dir -rnu uploads.txt

Searches the entire "DISK" all directories all levels for the file uploads.txt and displays if found in unformatted order. Note: the directories are appended to the file name as shown, unformatted operation only.

COM/UPLOADS/uploads.txt

chd /h0/sources/asm dir -n *.a

Displays all the files with the ".a" postfix in the SOURCES/ASM directory in un-sorted order. Note wild cards * ? only pertain to the current working directory specifying any other directory can produce interesting results.

Recursive Operation:

Up to 99 levels can be displayed recursively. Directories are opened in a first is last and last first order, so beware when using with other utilities such as "fbu". As an example I like my CMDS directory and it's contents on the lowest number tracks to reduce seek times, so by renaming the CMDS to ZMDS and using the "-w" option I can move the ZMDS directory to the head of the class. Upon restoration just rename to CMDS and apply the "-w" option.

Examples:

dir -r=1 /h0

Displays the contents of /h0 and the first level of directories such as the contents of CMDS, SYS etc.

dir -r=2 /h0

Displays the contents of /h0 and two levels of

AUSTRALIAN OS9 NEWSLETTER

directories such as CMDS, CMDS/ICONS, SYS, GAMES/SUB etc.

dir -rsu

Displays all the directories and sub directories of contained within the current "." directory unformatted and in sorted order.

Note: =<num> the "=<num>" is an optional parameter the default level is all.

Reorganizing Your Disk

The "-w" re-write your directory file in sorted order, it also removes all unused entries "deleted entries". Each entry requires 32 bytes, each time a file is added to a directory the directory file size increases by that amount. If you delete an entry the

32 bytes are not returned but instead the first letter of the file name is zeroed out. This zeroed entry then becomes available to the next file copied to that directory.

This option does not function with the -r or -n options.

Examples:

dir -xw

Organizes the CMDS directory in Alpha Numeric order permanently any subsequent files copied to the directory are added on to the end rather than any old place where a deleted entry was.

Bye for now.

oooooooooooo000000000oooooooooooo

More about the IBM - CoCo emulator

From: jeffv@physics.ubc.ca (Jeff Vavasour)
To: trs80@physics.ubc.ca
Subject: Future CoCo II emulator

This is just a quick note to the CoCo II emulator users on my list that I haven't forgotten about you. In fact, CoCo.EXE has been evolving over the past while. A short list of what you can expect (sometime a couple of months from now, most likely)...

Some of the features already written into Version 1.40:

- * SoundBlaster support with volume control
- * Game port joystick support
- * Read/write real CoCo disks from within the emulator (rather than having to use virtual disks, though they are also still available)
- * Improved recognition of artifacting modes
- * Supports non-standard graphics modes
- * Load snapshots at the MS-DOS prompt with "COCO snapname"
- * RETRIEVE, DSKINI, and emulator now support up to 80-track disks
- * Select the disk's side with RETRIEVE and DSKINI; 1.2Mb drive calibration bug now fixed
- * Keyboard layout customizer option
- * Improved printer support: 8-bit capable, adding LFs to CRs is now optional, and the printer output can be disabled
- * Cold restart option; warm reset now the more

convenient SHIFT-F10

- * A new option in the Options Menu to shell to the MS-DOS prompt without ending an emulator session
- * Selectable interrupt frequency: 50Hz or 60Hz
- * Optional ROM patch for Disk BASIC adds new commands LOADX, SAVEX, DIRX, LOADXM, SAVEXM, KILLX, DOSX to access files residing in your MS-DOS directory. (DOSX is equivalent to the Quit option.)
- * A new opcode which allows 6809-based programs to call 8086 code from within the emulator. (That's how the X commands access your MS-DOS directory.)

Other options I'd like to add:

- * Built-in file import/export menu
- * Ability to use Disk BASIC's DSKINI from within the emulator to format virtual or real disks
- * Ability to retrieve CoCo II ROMs out of a CoCo III. They're in there, but not normally accessible
- * ...CoCo III features??????

Actually, I may hold out some of the advanced disk features (e.g. direct access) for the shareware version, but I haven't decided yet. Anyway, just a note to let everyone know what's going on... Feel free to distribute this message if you think anyone else might be interested.

- Jeff

AUSTRALIAN OS9 NEWSLETTER

oooooooooooo000000000000oooooooooooo

Compressor extensions

Date: 01 Jul 94 05:08:11 EDT
From: SCHOFLD@mebv.mhs.comuserv.com
To: Bob Devries <bob@paxnet.com.au>
Subject: .GZ files -Reply

Here's the list of different suffixes;

arc	ARC archive	COMPRESSED
cpi	cpio archive	
cgz	cpio archive	GZIPped
cpz	cpio archive	COMPRESSED
dvi	DVI file	
dgz	DVI file	GZIPped
dvz	DVI file	COMPRESSED
doc	PTF README file	
gz	text/data	GZIPped
hqx	HQX archive	BINHEXed
iso	ISO9660 table	
lcv	Verbose cpio output list	
ltv	Verbose tar output list	
lzh	LZH archive	LZH compressed (eg lha)
lha	LHA archive	LZH compressed

```
ps PostScript file
eps Encapsulated PostScript file
pgz Postscript file GZIPped
egz EPS file GZIPped
psz PostScript file COMPRESSed
epz EPS file COMPRESSed
shr Shell archive
shz " " COMPRESSed
tar Tar archive
taz Tar archive COMPRESSed
tgz Tar archive GZIPped
( NOT gtz! This comes from a broken version of
GZIP )
txt ASCII text
txz ASCII text COMPRESSed
z,z Data/text file COMPRESSed
zip Data/text file PKZIPped
zoo ZOO archive ZOO'd ( eg zoozip )
```

There you go - there's a relatively limited number of utilities in use, it's just knowing the combination of letters to work out which utility to use!

oooooooooooo000000000000oooooooooooo

MM/1 Keysense function by Andrzej Kotanski

From: kotanski@13.75.DECNET.CERN.CH
Subject: reading MM/1 keyboard

Hello,

As the discussion about MM/1 keyboard driver has started again, I want to contribute to it - in a positive way. Here is the C function to check if certain important keys are pressed. Run the test program and see yourself. I apologise if certain

labels are not exact - my keyboard is not standard. But this contains all the information delivered by the keydrv - there is no data about the arrow keys, for instance.

Enjoy,
Andrzej Kotanski
(kotanski@zeus02.desy.de)

```
-----program starts here -----
/* ksysns.c - program to check certain MM/l keys - */  
/* released to PD by Andrzej Kotanski, April 24, 1994 */  
/* scroll lock is used to disable middle mouse button copying */  
/* run the test program after typing 'tmode nopause' */  
  
main()  
{  
    while (1)  
        ksysns();  
}  
  
char *wdata = 0;  
char *keydata;  
extern errno;  
kysns()  
{  
    if ( wdata != 0 )  
        wdata = 0;  
    if ( wd
```

AUSTRALIAN OS9 NEWSLETTER

```
    exit(errno);
keydata = wdata + 0xb4;
}

if ( keydata[2] & 1 )
    printf("left_shift ");
if ( keydata[2] & 2 )
    printf("right_shift ");
if ( keydata[3] )
    printf("scroll");
if ( keydata[4] )
    printf("control ");
if ( keydata[5] )
    printf("alt ");
if ( keydata[6] )
    printf("capslock ");
if ( keydata[7] )
    printf("numlock ");
if ( keydata[8] )
    printf("capslock2 ");
if ( keydata[9] )
    printf("numlock2 ");
if ( keydata[10] )
    printf("if_you_got_this_please_tell_the_author
");
    printf("\n");
}
```

Record locking with OS-9

From tripos!griffith@GOOFUS.WUSTL.EDU Tue Jun 28
07:45:10 1994
Date: Mon, 27 Jun 1994 11:57:19 CDT
From: Mark Griffith
<tripos!griffith@GOOFUS.WUSTL.EDU>
To: Multiple recipients of list COCO
<COCO@PUCC.PRINCETON.EDU>
Subject: Re: file and record locking

Mike K. sez:

>> I have been working on a rather unique database program under OSK 2.4, and wanted to make it "multi-user friendly", and would like to know what my options are for file and/or record locking. It's always been my understanding that OS-9, even 6809 L1, has file-locking, and maybe record locking, built into RBFMan, so you shouldn't need to set up external controls.

All too true....and a shame more people don't make use of it.

Basically, file locking is if you open a file in the UPDATE mode, no other user can also open it in UPDATE or WRITE mode, but can open it in READ mode. It's the same for record locking, where a record is OS-9 is any portion of a file delimited by the EOR (end-of-record) characters, in almost all cases, a carriage return. As Mike said, if you have a file opened in the UPDATE mode, other users can read any portion of that file EXCEPT that record your program has last read. To see an example of this, try downloading a GIF picture file from someplace and

running gifshow on it while it is downloading. Gifshow will move along reading the file in only as it is written out by your communications program. In fact, when Compuserve wrote the GIF specifications, they added in the ability to interlace the picture. Most people have seen this when they decode and display a GIF picture, it paints the screen three or four times, each one filling in more lines until the entire picture is there. This was put in to allow communications programs that display the pictures as they were being downloaded to give the user a good idea of what the picture would look like, and the opportunity to abort the transfer if they didn't like it. Under OS-9, we don't need something like this because the operating system allows you to basically do the same thing.

Now, to use the locking features in a database, all you really need to do is to open the file in the UPDATE mode and manipulate the file pointer. The record that the file pointer is currently at cannot be updated until the file pointer moves (I believe this is true....it may be that you only need to write out that record, which of course moves the file pointer). Now, if you make your database a binary file, then the OS can't use the ASCII end-of-record characters unless you keep that in there just for that purpose. Don't even think about changing your device descriptors to make a different EOR character. Forcing users to do that just to use your database is asking too much.

Mark

oooooooooooooooooooooooooooooooooooo

AUSTRALIAN OS9 NEWSLETTER

A discussion about the use of SCANF

From: Robert Gault <ab282@detroit.freenet.org>
Subject: fscanf() in C

I have been trying to use fscanf() as implemented in the Kreider 'C' library on this listserver. There are problems which maybe due to a bug in the Microware C, Kreider's clibt.l, or my understanding of the function. Help will be welcome.

The usage is: if (fscanf(infile,"%[^\\l\\n]",buff1) == EOF)
{
 body of routine
}

The above should read from a file up to but not including LF or CR. The file pointers should be updated so that the next pass through the routine reads more of the file. However, the same info is read each pass through the above.

This can't be correct! In the same usage, fgets() would work as desired with the exception that LF can't be handled. This indicates that all other sections of code are OK.

Any ideas?

From: James Jones <jejones@MICROWARE.COM>
Subject: Re: fscanf() in C

In article <1
99406121354.AA00377@mcrware.microware.com>
ab282@detroit.freenet.org writes:
>The usage is: if (fscanf(infile,"%[^\\l\\n]",buff1) == EOF)
>{
> body of routine
>}

fscanf() returns the number of items converted, so you might want to write that as "fscanf(...)==1".

>The above should read from a file up to but not including LF or CR. The file pointers should be updated so that the next pass through the routine reads more of the file.

Well...fscanf() has to read the LF or CR to know when to stop, but it will ungetc() it, which may make it give you the empty string in buff1 next time through. You might have better luck if you make it grab the CR or LF, so:

```
if (fscanf(infile, "%[^\\l\\n]c", buff1,  
&junk) == 2) {  
    /* process buff1 */  
}  
(with "char junk;" declared at the right spot earlier  
on).
```

James Jones

Opinions (and mistakes) herein are those of the author, and not necessarily those of any organization.

From: Zach Heilig <heilig@CS.UND.NODAK.EDU>
Subject: Re: fscanf() in C

In article <9406121044.AA03026@detroit.freenet.org>,
Robert Gault <ab282@detroit.freenet.org> wrote:
>The usage is: if (fscanf(infile,"%[^\\l\\n]",buff1) == EOF)

```
>{  
>    body of routine  
>}
```

>The above should read from a file up to but not including LF or CR. The file pointers should be updated so that the next pass through the routine reads more of the file. However, the same info is read each pass through the above.

Well, one possible problem is:
'\\n' == LF (or new-line)
'\\r' == CR.

-- Zach Heilig (heilig@cs.und.nodak.edu)
--

(heilig@agassiz.cas.und.nodak.edu)

From: Robert Gault <ab282@detroit.freenet.org>
Subject: fscanf()

Thankyou to all who replied to my question. I might as well tell you why I asked. In our list server, we have CUTS for OS-9. This program is not robust enough in its handling of line termination; requiring CR and bombing on LF.

This could be handled by writing a seperate program to convert all LF to CR, and some "smart" terminal programs may permit selection of the EOL character. I feel the esthetic solution is to modify cuts_decode.c

So far this has meant replacing an fgets() with the

AUSTRALIAN OS9 NEWSLETTER

fscanf(*file, [^\l\n],*buff) and adding a routine to tack a CR on the end of buff plus getc() the next character (LF or CR) from the file.

Everything seems to be working except when I have two LF or CR back to back; ie. LF LF. In this case the following routine has problems and spits out a duplicate buff.

```
while (('\l'  '\n') == getc(infp));
```

The above "should" be able to eat any number of consecutive LFs but it hasn't worked so far.

From: James Jones <jejones@MICROWARE.COM>
Subject: Re: fscanf()

In article <1
99406131056.AA15145@mcrware.microware.com>
ab282@detroit.freenet.org writes:
>
>while (('\l' '\n') == getc(infp));
>
>The above "should" be able to eat any number of consecutive LFs but it hasn't worked so far.

Oops. I think something like that worked in COBOL, but in C, that's comparing getc(infp) with l, because that's the result of the expression '\l' '\n'. Because of the side effects of getc(), you'd either have to call a function or use a variable to hold the result:

```
while ((n = getc(infp)) == '\l'  n == '\n')  
;
```

James Jones

Opinions herein are those of their respective authors, and not necessarily those of any organization.

From: knudsen@IHADES.ATT.COM
Subject: Re: fscanf() in C

Some comments on my own and others' replies:

>The usage is: if (fscanf(infile,"%[^l\n]",buff1) == EOF)
>Hmmm, what's that 3rd "buff" arg for? That's for sscanf(),
The 3rd (and any more) args would be for percent (%) constructs to put their "catches" into. The example does sort of have a % construct -- apparently

%[^ ...]

means to recognize any one of the characters inside the brackets? I've never heard of that one before, not even in the newer C books that came out when C got popular under MSDOS.

I'm certain it would not be recognized in the Microware 6809 C library, though it may have gotten into Carl Kreider's latest (?) I won't even bother testing the AT&T C compiler libs...

Seems every few months another printf/scanf format option turns up...

James Jones sez:

>Well...fscanf() has to read the LF or CR to know when to stop, but it will ungetc() it, which may make it give you the empty string in buff1 next time through.

This is the most elegant explanation I've yet seen for the weird behavior of scanf() family fcns when you put an explicit whitespace delimiter in your format string. Hmmm, it doesn't really jam or trash the input system, just puts the control character back. Nice to finally know why you shouldn't put explicit newlines in the string...

> You might have better luck if you make it grab the CR or LF, so:
> if (fscanf(infile, "%[^l\n%c", buff1,
&junk) == 2) {
> /* process buff1 */
> }

Lessee, the first (funny) % construct scans up to and recognizes the newline, then the 2nd one (%c) eats and grabs it for keeps. Boy, talk about BASIC/FORTRAN spaghetti code...but given my remarks about the ubiquity of stdio's bugs and features, this code may even be portable.

You could use a getchar() or fgetc() to get the byte instead of the %c, but James' version is easier to retrofit into the Cuts.c code.

From: knudsen@IHADES.ATT.COM
Subject: Re: fscanf() in C

>I have been trying to use fscanf() as implemented in the Kreider 'C' library on this listserver. There are problems which maybe due to a bug in the Microware C, Kreider's clibt.l, or my understanding of the function. Help will be welcome.

AUSTRALIAN OS9 NEWSLETTER

> This can't be correct!

But it is "correct", unforch....

Actually, the scanf() family of I/O functions has had this same bug/misfeature since the days when Kernighan and Ritchie were lowly grunt engineers at Bell Labs. The misfeature has been dutifully copied from UNIX into Microware's library, and apparently into Carl's as well. (Trying to fix it would probably break compatibility).

>The usage is: if (fscanf(infile,"%[^\\l\\n]",buff1) == EOF)
> {
> body of routine
> }

Hmmm, what's that 3rd "buff" arg for? That's for sscanf(), where you've already read something into the buffer, like with fgets(). Maybe that 3rd arg is confusing the function.

Anyway, the best rule is: NEVER put a \l, \n, or \r in a scanf() string! The scanf won't get past it (as you found) and may even go into the famous tailspin whereby the last stuff read gets blown out on stdout, over and over (you missed that, shucks :-)

I think scanf() automatically takes newlines and such as delimiters between fields of numbers and strings, and so gets confused when you mention these explicitly. Like, it doesn't know how to find something it's been designed to ignore (?)

Another good rule is: Except for quickie throwaway programs, NEVER read disk files directly with fscanf(). Instead, read each line into a buffer and use sscanf(buff, ...) on it. This does pose problems when the input is broken over several lines; then fscanf() really is good in that it will ignore the linefeeds and space over them.

>The above should read from a file up to but not including LF or CR.

Actually, the intent of fscanf() is to read fields of data while ignoring line breaks. If you want linefeeds to be significant as delimiters, use fgets() to get per-line input (this is C's version of LINE INPUT) and then analyze the line with sscanf().

If you really need "full control over the dirt" then read chars one at a time with fgetc(), looking for white-spacers and copying the other bytes into a

buffer. Then use sscanf() to break out the data fields in the string buffer.

Sorry this is a little incoherent, but that matches the subject matter. --mike k

From: Zach Heilig <heilig@CS.UND.NODAK.EDU>
Subject: Re: fscanf() in C

In article <9406142235.AA02149@igl.att.att.com>, <knudsen@IHADES.ATT.COM> wrote:
>The usage is: if (fscanf(infile,"%[^\\l\\n]",buff1) == EOF)
>
>>Hmmm, what's that 3rd "buff" arg for? That's for sscanf(), The 3rd (and any more) args would be for percent (%) constructs to put their "catches" into.
>The example does sort of have a % construct -- apparently
> %[^ ...]
>means to recognize any one of the characters inside the brackets? I've never heard of that one before, not even in the newer C books that came out when C got popular under MSDOS. I'm certain it would not be recognized in the Microware 6809 C library, though it may have gotten into Carl Kreider's latest (?) I won't even bother testing the AT&T C compiler libs...

Hmm. I think I'll stick my head out yet again (after proving some ignorance about the '\l', '\n', '\r' stuff earlier :-)

char foo[];
scanf("%[x]", foo);
would copy all the x's that are next to each other into the array.

scanf("%[a-z]", foo);
would copy a whole sequence of only lower case letters into the array.

scanf("%[^a]", foo);
would copy all the characters up till (but not including) a lower case 'a' into the array.

scanf("%20[^]", foo);
would copy at most 20 (or is that 19 + a null char) characters or up till (but not including) the first ']' into the array.

if the libraries support this or not, I'm not sure (I suspect that they do). -- Zach Heilig (heilig@cs.und.nodak.edu) == (heilig@agassiz.cas.und.nodak.edu)

AUSTRALIAN OS9 NEWSLETTER

oooooooooooo0000000000oooooooooooo

Some thoughts about copyright discussed on the Internet CoCo List

From: Allen Huffman <coco-sysop@GENIE.GEIS.COM>
Subject: Dangers of Orphanware

I just read something in the UpTime editorial that really bothers me.

There had been people writing asking where to obtain the Planet Engine program for OS-9. No one seemed to know who was carrying it since the company that advertised it in Rainbow had long since disappeared. Well, we who have gone to the 'Fests these past years KNOW it's still active, alive, and well - sold by Lee Veal and David Wordell of the Dallas Tandy/Radio Shack Users Group (DALTRUG). Lee and David have made most all of the past 'Fests, including the Middle America Fest in Iowa.

Well, the UpTime article states "Recently, a few users have asked where they may obtain the Planet Engine program from Lee Veal. We could not contact Lee directly or indirectly, and at the time I received some information that said the MI&CC club was distributing that as part of their library."

For those who don't know, MI&CC is the Mid Iowa & County CoCo Club. They are a strong organization with a massive library of "orphanware" products. Most of these products can be dated back to the classic and gone CoCo 1 and 2 days - liked to companies that are no longer anywhere in the market.

Now, even though it's a violation of copyright law, it's difficult to see the harm in obtaining a 10 year old program that NO ONE is offering any more. Thus, it is justified with the statement that they will gladly remove any program if they hear from the copyright owner.

Well --- there is a problem here. It is something I thought about when I went to the MidAmFest and found out about their library. I thought "what if they had a program in their library that was no longer available and then later someone started to revive it and bring it back to market? would anyone be interested at that point, seeing it freely float around to possibly hundreds of CoCoers through that library?"

Well this seems to have happened. If it is true (I'd like confirmation on this) that the MI&CC club DID

indeed have Planet Engine in their library for \$3 a disk, how has that affected the legitimate sale of the program by DALTRUG? ("The program was written by and is still owned by Paul Light. He has licensed DalTrug, our local club, as the sole distributor of this product." writes David Wordell in UpTime.)

Ladies and gentlemen, this is not a good situation. When Sub-Etha Software disappeared from the pages of Rainbow and seemed to "vanish" from the masses, we still appeared at all the Atlanta and ChicagoFests every year. It makes me wonder if one of my programs might have ended up on their club library. Ditto on many other vendors that are still around, just not advertising. Eversoft is still there (though not as active) - but is Crystal Quest or the 3D PacMan game sitting on some orphanware disk somewhere?

Letting a program out like this cheapens its values. It's easier to see spending \$20 for a program you think you want if you haven't already "paid" for it from a club.

I'd like to hear some comments on this. I, for one, really think that if clubs wish to continue the practice of orphanware, they need to set some kind of limitation on what they accept. ie, "no programs made within the last seven years" or something.

Allen C. Huffman -/- Sub-Etha Software
(Lufkin, TX)

From: Mark Griffith
<tripos!griffith@GOOFUS.WUSTL.EDU>
Subject: Re: Dangers of Orphanware

I think the copyright laws, that are designed to protect the authors, should be the final authority in these cases. I'm not a copyright lawyer (and I never played on TV), but my understanding is a copyrighted work is protected for at least seven years (possibly more), but only if the copyright owner has filed his copyright with the government.

This is not to say that the owner can't protest and sue someone infringing on his copyright, just that without filing, it is very difficult to prove he/she had the copyright when they claim they did.

A club or some other organization should not sell

AUSTRALIAN OS9 NEWSLETTER

copyrighted works unless they have clear documentation that the work has been placed in the public domain, or the copyright protection limits have expired, which places the work in the public domain.

The bottom line is this: if a club wants to put something on their disks that still has a valid copyright they do so at their own risk. The copyright owner can sue them and will probably win, IF the copyright was registered. The club is taking the chance.

In the CoCo world, I seriously doubt anyone that wrote a program for the CoCo and then moved on to the bigger world would care. It may be illegal or at least immoral for the club to do that, but only the copyright owner can do anything about it.

In your case, of you have clear documentation that the copyright owner has licensed the product to you for sale, then you need to contact the club, in writing, and tell them you are expecting to receive the full sum that you sell the product for from them for every copy they have sold. If they don't comply, sue them. If it were me, I'd say something like:

"It is my understanding that you have sold 100 copies of club disks, and each disk contained a copy of "The Planet Engine" software. Since we are the sole licensed agency to sell "The Planet Engine", and our selling price is \$29.95, we expect to receive a check from you for the amount of \$2995.00 within 90 days. If this amount is not received by that time, we will be filing a suit in the <blank> county court against you for grand larceny."

That should at least get their attention.

/\\ark

From: knudsen@IHADES.ATT.COM
Subject: Re: Dangers of Orphanware

[Allen Huffman's text deleted]

Well, this sort of question puts me in a quandary. I've always advocated the right of Coco users to copy and "pirate" software that is clearly out of print and no longer available, especially Tandy utilities and such.

But I have to be careful, since the performance of UltiMusE dealers sometimes leads folks to wonder if UltiMusE isn't an orphaned product, and next thing you

know my program will be floating around under the Jolly Roger.

The answer to me seems not to be a fixed-time waiting period, but some better detective work and "devil's advocacy" before any club or person declares that a program is OOP (outa print) and therefore OK to copy. At least, post queries to this List, CIS, and Delphi about "anyone selling this thing?" And "will the author please contact so-and-so about this?"

As for someone "reviving" an orphaned program and starting to sell it again after a long hiatus -- well, that brings up the rights of the original author, and whether he still owns the rights, and cares.

Even tho nobody is selling or distributing a program any more, you still need to consider the author's wishes. If you can find him.

It's true that Rainbow's outa-sight ad prices drove a lot of vendors "underground" or "dark" and could have given people the wrong ideas. Also a few authors, like the writer of Shell+, don't get on any electronic forums at all and are anonymous, so it's hard to check with the rightful owner. (Since Shell+ is freeware that's a bad example, but you see what I mean.)

I do have to wonder how a Michigan club could miss the last Chicago CocoFest, where Planet Engine was very visibly being demonstrated and sold at a booth near the door!

From: "MSgt Timothy Daugard, XPOL, 4-4837"
<tdaugard@hosea.afsoc.af.mil>
Subject: Re: Dangers of Orphanware

Mark Griffith wrote:

>> ... the copyright laws, that are designed to protect the authors, should be the final authority in these cases. I'm not a copyright lawyer, but my understanding is a copyrighted work is protected for at least seven years (possibly more), but only if the copyright owner has filed his copyright with the government.

I'm not a copyright lawyer either, but have done some research support of a career in art. Current law (changed late sixties - early seventies (I don't have my books here at work)) protects copyrighted work for the life of the author. For "works for hire" I believe the protection runs 50-70 years. Protection

AUSTRALIAN OS9 NEWSLETTER

is renewable once on the death of the author i.e. Elvis, Inc. This copyright is valid whether the work has been filed or not.

If the work has been filed and granted a copywrite notice the author can collect actual damages, punitive damages and court costs. If the work is unregistered the author can only collect actual damages (and maybe court costs?).

>> This is not to say that the owner can't protest and sue someone infringing on his copyright, just that without filing, it is very difficult to prove he/she had the copyright when they claim they did.

Copyright goes in effect the moment the work is published, even if the work is not correctly marked. Courts have ruled that lack of copyright markings does not void the authors rights. I deal mostly with paintings and prints but I believe advertising in print such as rainbow would be sufficient proof. The rainbow seal should be considered good proof also.

>> A club or some other organization should not sell copyrighted works unless they have clear documentation that the work has been placed in the public domain, or the copyright protection limits have expired, which places the work in the public domain.

Given the age of the computer revolution, there is no software for personal computers in existence with an expired copyright. Some software written in the fifties and forties for main frames may not be copyrighted (written prior to current law)

Any club selling software must ensure the software is in the public domain or have at least limited rights to the copyrighted software granted by the copyright holder. Public domain software can only be transmitted under the conditions stated by the author when it is placed in the public domain.

>> The bottom line is this: if a club wants to put something on their disks that still has a valid copyright they do so at their own risk.

Agree completely

>> copyright owner can sue them and will probably win, IF the copyright was registered. The club is taking the chance.

Depending on how the club is set up, not only is the club taking the chance, all members of the club could

be fiscally responsible.

>> In the CoCo world, I seriously doubt anyone that wrote a program for the CoCo and then moved on to the bigger world would care. It may be illegal or at least immoral for the club to do that, but only the copyright owner can do anything about it.

Agree

>> In your case,If they don't comply, sue them. If it were me, I'd say something like: ... we expect to receive a check from you for the amount of \$2995.00 within 90 days. If this amount is not received by that time, we will be filing a suit in the <blank> county court against you for grand larceny."

Talk to a lawyer, I believe it would be "...against you for copyright infringement."

An aside: Some countries of the world have not signed the international copyright agreement (for a long time the US was one of them). It is legal to copy software in those countries - but the software cannot be sold or imported to any country that has signed the agreements.

As I said before, I disavow any responsibility for the legal content of this, for legal advice consult a lawyer.

Tim Daugard

tdaugard@libra.afsoc.af.mil

From: Mark Griffith
<tripos!griffith@GOOFUS.WUSTL.EDU>
Subject: Re: Dangers of Orphanware

Timothy Daugard sez:

> If the work has been filed and granted a copywrite notice the author can collect actual damages, punitive damages and court costs. If the work is unregistered the author can only collect actual damages (and maybe court costs?). Copyright goes in effect the moment the work is published, even if the work is not correctly marked. Courts have ruled that lack of copyright markings does not void the authors rights. I deal mostly with paintings and prints but I believe advertising in print such as rainbow would be sufficient proof. The rainbow seal should be considered good proof also.

AUSTRALIAN OS9 NEWSLETTER

I believe this is all correct, although some lawyer would have to sort out the details. I do think that although the work may be protected by copyright laws the moment it is published, it would still be harder to make an infringement claim in court if it wasn't registered. I remember this from some copyright materials I got from the US gov. The point is, if you want to go all the way with your software (marketing it as a product), you'd better register it.

> Given the age of the computer revolution, there is no software for personal computers in existence with an expired copyright.

I'm in complete agreement here. However, there is software that was previously copyrighted that the author has subsequently placed in the public domain. Clubs and users need to be aware of this since this software many times will still show the copyright message on the startup screen.

In Allen Huffman's case, I'd go after the buggers and teach them a lesson they won't soon forget. I might not force them to pay, but I'd at least make them sweat for a while.

The point is that no matter how you look at it, there are specific laws that are designed to prevent this "orphanware" problem and they should be followed.

^vark

From: David Halko <halko@scuzzy.attmail.com>
Subject: Orphanware

Hello Folks...

I think it is up to the supporter of the Orphanware product to find the author and ask permission to release it...

This orphanware product should also TRY TO GET THE SOURCE so the program can do some good for the community. With the source floating around, people can learn about programming, animation, etc.

I have been considering for some time going through the process of finding OS-9 orphanware products and re-releasing them again. It is an easy thing to do as soon as you pick up a few years worth of rainbow and read their advertisements.

Dave Halko

From: Allen Huffman <coco-sysop@GENIE.GEIS.COM>

Subject: Re: Dangers of Orphanware

>[Allen Huffman's text deleted]

<hmph> ;-)

>The answer to me seems not to be a fixed-time waiting period, but some better detective work and "devil's advocacy" before any club or person declares that a program is OOP (outa print) and therefore OK to copy. At least, post queries to this List, CIS, and Delphi about "anyone selling this thing?" And "Will the author please contact so-and-so about this?"

That would be easy enough. Heck, if they were selling Planet Engine, they must not have read any of my 'Fest reports from the last three or so years. You know, Lee Veal and David Wordell were actually AT the Middle America Fest as vendors selling it!

> As for someone "reviving" an orphaned program and starting to sell it again after a long hiatus -- well, that brings up the rights of the original author, and whether he still owns the rights, and cares.

This would, of course, be done with the original author's permission. For instance, remember Pyramix? Rumour has it that the classic Q-Bert CoCo 3 game might return to the marketplace. The author is still active and doing things on a monthly basis --- but that particular program just hasn't been around. Is it on one of their disks?

> Even tho nobody is selling or distributing a program any more, you still need to consider the author's wishes. If you can find him.

YES. Period.

>It's true that Rainbow's outa-sight ad prices drove a lot of vendors "underground" or "dark" and could have given people the wrong ideas. Also a few authors, like the writer of Shell+, don't get on any electronic forums at all and are anonymous, so it's hard to check with the rightful owner. (Since Shell+ is freeware that's a bad example, but you see what I mean.)

Efforts need to be made, that is for sure, to ensure that the program isn't out there somewhere. Instead, they seem to have a "we'll do it, and it's up to the owner to complain" which is a side I can see --- at least that points out they aren't intentionally trying to swap someone's sales.

AUSTRALIAN OS9 NEWSLETTER

> I do have to wonder how a Michigan club could miss the last Chicago CocoFest, where Planet Engine was very visibly being demonstrated and sold at a booth near the door!

Actually, MI&CC is in Iowa. Middle-Iowa & Country CoCo.

Allen C. Huffman -/- Sub-Etha Software
(Lufkin, TX)

00000000000000000000000000000000

IIIIII	NNNN	NN	FFFFFF	000000	!!
II	NN NN	NN	FF	00 00	!!
II	NN NN	NN	FFFF	00 00	!!
II	NN	NN NN	FF	00 00	!!
II	NN	NN NN	FF	00 00	!!
IIIIII	NN	NNNN	FF	000000	00

By Rod Holden

Hi, and welcome to Info. This particular piece of software is in a file called ded3.pak which is available under the OCN OS9_UTI directory. This is not the complete document as is was to big for the newsletter, please read on;

dEd
Disk Editor
Copyright 1987 Doug DeMartinis

dd is a screen-oriented disk editor utility written in assembly language. It was originally conceived as a floppy disk editor, so the display is organized around individual sectors. It performs most of the functions of Patch, from Computerware, but is faster, more compact, and screen-oriented rather than line-oriented. Individual files or the disk itself (hard, floppy, ram) can be examined and changed, sectors can be written to an output file, and executable modules can be located, linked to and verified. With simple changes, it will run on any CoCo Level I OS-9 and possibly others (CoCo Level II OS-9).

To use, type:

dEd <pathlist>

where <pathlist> is of the form:
filename or dirname
or /path/filename
or /DOS (edits entire disk)

dEd will read in and display the first 256 bytes in the file (disk). This is Logical Sector Number (LSN) zero. You move through the file sector (LSN) by sector using the up and down arrow keys. The current LSN number is displayed in Hex and Decimal in the upper left corner of the screen. If the disk itself was accessed (by appending '@' to it's name when dEd was called), the LSN is the disk sector number. If an individual file is being edited, however, the LSN displayed refers to the file, not to the disk. All numbers requested by dEd must be in Hex format. All commands are accessed by simply pressing the desired key.

Bye for now.

ooooooooooooooo000000000000oooooooooooo